Listing of Claims:

1. (currently amended) A computerized method for updating a version of an object having a property, the method comprising:

receiving an updated value for the property~~, wherein the property is a piece of data of the object~~;

setting an end version field in a first data structure ~~an object table of an object repository or database~~ to a value representing a predecessor version of the object; and

creating a second data structure by: ~~object table in the object repository or database to represent a successor version of the object by:~~

setting a start version field in the second data structure ~~object table~~ to a value representing the successor version of the object;

setting an end version field in the second data structure ~~object table~~ to a value representing a most recent version of the object; and

setting a property value field in the second data structure to the updated value for the property, wherein the start version field and the end version field in the second data structure define a range of versions including the updated value for the property; ~~for which the value of the property has the same value.~~

wherein version and property value fields of the data structures record properties of the object and associated versions of the object facilitating a recalling and generating of the object without requiring a copying of the object.

2. (canceled)

3. (previously presented) The computerized method of claim 1, wherein the value representing the most recent version is infinity.

4. (original) The computerized method of claim 1, wherein the data structure is a row in a database.

5. (original) The computerized method of claim 1, wherein the object is a COM (Component Object Model) object.

6. (currently amended) A computer-readable medium having a data structure ~~an object table for maintaining multiple versions of an object~~ stored thereon, the medium comprising:

a first field comprising a key identifying an object;

a second field comprising a start version identifier;

a third field comprising an end version identifier; and

a fourth field comprising a property value ~~identifying at least one piece of data~~ of the object; ~~and~~

wherein the second and third field define a range of versions of ~~an~~ the object identified by the first field having the property value in the fourth field. and;

wherein a plurality of data structures define properties of the object and record associated versions of the object allowing for recall and generation of the object without requiring a copying the object.

7. (original) The computer-readable medium of claim 6, wherein the first field comprises an object identifier and a branch identifier.

8. (currently amended) A computer-readable medium having computer-executable instructions for updating a version of an object having a property, the method comprising:

receiving an updated value for the property~~, wherein the property is a piece of data of the object~~;

setting an end version field in a first data structure ~~object table of an object repository or database to~~ a value representing a predecessor version of the object; and

creating a second data structure by: ~~object table in the object repository or database to represent a successor version of the object by:~~

setting a start version field in the second data structure ~~object table~~ to a value representing the successor version of the object;

setting an end version field in the second data structure ~~object table~~ to a value representing a most recent version of the object; and

setting a property value field in the second data structure to the updated value for the property, wherein the start version field and the end version field in the second data structure define a range of versions including the updated value for the property; ~~for which the value of the property has the same value.~~

wherein version and property value fields of the data structures record properties of the object and associated versions of the object facilitating a recalling and generating of the object without requiring a copying of the object.

9. (canceled)

10. (original) The computer-readable medium of claim 8, wherein the value representing the most recent value is infinity.

11. (canceled)

12. (original) The computer-readable medium of claim 8, wherein the object is a COM (Component Object Model) object.

13. (currently amended) A method for propagating a relationship of a predecessor object to a successor object, said relationship having an origin object and a destination object, the method comprising:

reading a propagation flag on the relationship; and

if the propagation flag is set then performing the tasks of:

determining if a previously added version of the destination object has been added;

upon determining the previously added version has been added:

setting an end version field in a first data structure ~~an object table of an object repository or database~~ with a value representing a predecessor version of the object;

creating a second data structure ~~object table in the object repository or database~~ to represent a successor version of the object by:

setting a start version in the second data structure ~~object table~~ to a value representing the successor version;

setting an end version field in the second data structure ~~object table~~ to a value representing a most recent version of the object; and

setting a property value field in the second data structure to the updated value for the property, wherein the start version field and the end version field in the second data structure define a range of versions including the updated value for the property; ~~for which the value of the property has the same value.~~

wherein version and property value fields of the data structures record properties of the object and associated versions of the object facilitating a recalling and generating of the object without requiring a copying of the object.

14. (canceled)

15. (currently amended) A computer-readable medium having computer executable instructions for performing a method for propagating a relationship of a predecessor object to a successor object, said relationship having an origin object and a destination object, the method comprising:

reading a propagation flag on the relationship; and

if the propagation flag is set then performing the tasks of:

determining if a previously added version of the destination object has been added;

upon determining the previously added version has been added:

setting an end version field in a first data structure an object table of an object repository or database with a value representing a predecessor version of the object;

creating a second data structureby: ~~object table in the object repository or database to represent a successor version of the object by;~~

setting a start version in the second data structure ~~object table~~ to a value representing the successor version;

setting an end version field in the second data structure ~~object table~~ to a value representing a most recent version of the object; and

setting a property value field in the second data structure to the updated value for the property, wherein the start version field and the end version field in the second data structure define a range of versions including the updated value for the property; ~~for which the value of the property has the same value.~~

wherein version and property value fields of the data structures record properties of the object and associated versions of the object facilitating a recalling and generating of the object without requiring a copying of the object.

16 - 37. (canceled).

38. (currently amended) The computer-readable medium of claim 6, wherein the data structure maintains property values for multiple versions of the object and wherein the object is not copied ~~objects and properties are only copied to the object table data structure~~ when a property value of the ~~a respective~~ object changes.

39. (currently amended) The computer-readable medium of claim 38 ~~6~~, wherein the first field includes an object identifier and~~,~~ a branch identifier. ~~, and a start-version identifier.~~

40 - 41. (canceled)

42. (previously presented) The computer-readable medium of claim 39, wherein the branch identifier indicates a branch within a particular version of the object, the branch being formed when a previously added successor object is created from a predecessor object having at least one other successor object.

43. (canceled)

44. (previously presented) The method of claim 13, wherein, if the propagation flag is set, the relationship is not copied to the previously added version.

45. (previously presented) The method of claim 13, wherein reading a propagation flag on the relationship involves reading a relationship type field of a relationship table, the relationship table including an object identifier, a branch identifier, a start-version identifier, and an end-version identifier.

46. (previously presented) The method of claim 45, wherein, when creating the previously added version, if the previously added version and a predecessor version are on the same branch, as indicated by the branch identifier, and the end-version identifier is infinity, the relationship is copied without updating the relationship table.

47. (previously presented) The method of claim 45, wherein a previously added row of the relationship table is created when a previously added branch is created, as indicated by the branch identifier.

48. (previously presented) The computer-readable medium of claim 15, wherein, if the propagation flag is set, the relationship is not copied to the previously added version.

49. (previously presented) The computer-readable medium of claim 15, wherein reading a propagation flag on the relationship involves reading a relationship type field of a relationship table, the relationship table including an object identifier, a branch identifier, a start-version identifier, and an end-version identifier.

50. (previously presented) The computer-readable medium of claim 49, wherein, when creating a previously added version, if the previously added version and a predecessor version are on the same branch, as indicated by the branch identifier, and the end-version identifier is infinity, a relationship is copied without updating the relationship table.

51. (previously presented) The computer-readable medium of claim 49, wherein a previously added row of the relationship table is created when a previously added branch is created, as indicated by the branch identifier.

52. (new) The method of claim 1, wherein a version of an object having a property is updated and recorded without copying the updated object.

53. (new) The method of claim 1, wherein a version of an object having a property is updated by copying only updated properties and associated version identifiers and not the updated object.

54. (new) The method of claim 1, wherein the property is a name-value pair and wherein the name refers to and performs operations on the value.

55. (new) The method of claim 1, wherein the method creates data structure only for properties that have changed value and does not copy the updated object.

56. (new) The method of claim 1, wherein the method creates or modifies property value fields only for properties that have changed value and creates or modifies version fields for only the versions including the properties that have changed value.

57. (new) A method for changing a value of a property of an object, the method comprising:

receiving a value for a property of an object;

setting a first end version to a first end value representing a version of the object, wherein a first start version and the first end version are associated with a predecessor value of the property, wherein the first start version has a first start value, wherein the first start value of the first start version and the first end value of the first end version define a first range of versions of the object for which the property has the predecessor value;

setting the property to the value for the property received;

setting a second start version to a second start value representing a version of the object, wherein the second start version is associated with the value for the property received, wherein the second start value of the second start version defines a start of a second

range of versions of the object for which the property has the value for the property received; and

setting a second end version to a second end value representing a version of the object, wherein the second end version is associated with the value for the property received, wherein the second end value of the second end version defines an end of the second range of versions of the object for which the property has the value for the property received;

wherein the object including the value of the property received can be generated without copying the object including the value of the property received.

58. (new) The method of claim 57, wherein the object can be used or operated through one or more interfaces, wherein each of the one or more interfaces refers to a specification for any number of properties or any number of methods or any number of behaviors.

59. (new) The method of claim 57, wherein the value of the property received is stored in an object repository or a database, wherein the object repository or database uses any number of data structures to define the object but does not store the object.

60. (new) The method of claim 57, wherein the first start version is a first start version field of a first data structure, wherein the first end version is a first end version field of the first data structure, wherein the first data structure is associated with the predecessor value of the property.

61. (new) The method of claim 60, wherein the first data structure is represented by one or more rows of one or more tables, wherein the first start version field is a first field of the one or more rows, wherein the first end version field is a second field of the one or

more rows, wherein the predecessor value of the property is a third field of the one or more rows.